

基于拟牛顿法的深度强化学习在车联网边缘计算中的研究

章坚武¹, 芦泽韬¹, 章谦骅^{2,3}, 詹明⁴

(1. 杭州电子科技大学通信工程学院, 浙江 杭州 310018; 2. 之江实验室天基计算研究中心, 浙江 杭州 311121;
3. 浙江大学信息与电子工程学院, 浙江 杭州 310027; 4. 台州学院电子与信息工程学院, 浙江 台州 318000)

摘要: 为了解决车联网中由于多任务和资源限制导致的任务卸载决策不理想的问题, 提出了拟牛顿法的深度强化学习双阶段在线卸载 (QNRLO) 算法。该算法首先引入批归一化技术优化深度神经网络的训练过程, 随后采用拟牛顿法进行优化, 有效逼近最优解。通过此双阶段优化, 算法显著提升了在多任务和动态无线信道条件下的性能, 提高了计算效率。通过引入拉格朗日算子和重构的对偶函数, 将非凸优化问题转化为对偶函数的凸优化问题, 确保算法的全局最优性。此外, 算法考虑了车联网模型中的系统传输时间分配, 增强了模型的实用性。与现有算法相比, 所提算法显著提高了任务卸载的收敛性和稳定性, 并能有效处理车联网中的任务卸载问题, 具有较高的实用性和可靠性。

关键词: 车联网; 任务卸载; 深度强化学习; 拟牛顿法

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024101

Research on deep reinforcement learning in Internet of vehicles edge computing based on Quasi-Newton method

ZHANG Jianwu¹, LU Zetao¹, ZHANG Qianhua^{2,3}, ZHAN Ming⁴

1. School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

2. Research Center for Space Computing System, Zhejiang Lab, Hangzhou 311121, China

3. College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China

4. College of Electronic and Information Engineering, Taizhou University, Taizhou 318000, China

Abstract: To address the issues of ineffective task offloading decisions caused by multitasking and resource constraints in vehicular networks, the Quasi-Newton method deep reinforcement learning dual-phase online offloading (QNRLO) algorithm was proposed. The algorithm was designed by initially incorporating batch normalization techniques to optimize the training process of deep neural networks. Subsequently, optimization was performed using the Quasi-Newton method to effectively approximate the optimal solution. Through this dual-stage optimization, performance was significantly enhanced under conditions of multitasking and dynamic wireless channels, improving computational efficiency. By introducing Lagrange multipliers and a reconstructed dual function, the non-convex optimization problem was transformed into a convex optimization problem of the dual function, ensuring the global optimality of the algorithm. Additionally, system transmission time allocation in the vehicular network model was considered, enhancing the practicality of the algorithm. Compared to existing algorithms, the proposed algorithm improves the convergence and stability of task offloading significantly, addresses task offloading issues in vehicular networks effectively, and offers high practicality and reliability.

Keywords: Internet of vehicles, task offloading, deep reinforcement learning, Quasi-Newton method

收稿日期: 2024-02-04; 修回日期: 2024-05-07

通信作者: 章谦骅, zhangqh@zhejianglab.com

基金项目: 浙江省自然科学基金重点项目 (No.LZ23F010001)

Foundation Item: Key Program of Zhejiang Provincial Natural Science Foundation (No.LZ23F010001)

0 引言

在过去的二十年中,云计算取得了显著的发展,推动了各种应用程序和服务的兴起^[1]。这些应用程序和服务能够在极大的规模上运行,支撑着各种不同的云平台。这些云平台利用最新的虚拟化技术和资源管理机制,按需提供大量的虚拟资源。然而,随着时间的推移,为了满足时延敏感的新型应用需求,物理资源的中心化趋势逐渐向边缘化转移,进而催生了如边缘计算、雾计算等新的计算范式^[2-5]。

在这样的背景下,车载自组织网络(VANET, vehicular ad-hoc network)逐渐成为研究的热点^[6]。车联网(IoV, Internet of vehicles)通过智能设备和无线通信技术,使车辆能够相互通信,从而提高道路安全性和交通效率^[7]。由于车辆数量的不断增加和交通流量的不对称分布,网络运营商必须设计智能卸载策略以提高网络性能,并为用户提供高质量的服务。然而,全局信息缺乏和车联网的时间多样性使得在路侧单元(RSU, road side unit)的长期能源限制下进行有效的卸载和缓存决策变得具有挑战性^[8]。

近年来,已有许多研究围绕车联网边缘计算的任务卸载问题展开。文献[9]提出了一种自学习框架,以根据车辆的计算需求和边缘服务器的可用资源来自适应地选择最佳边缘服务器。文献[10]提出了一种新的边缘计算任务调度算法,基于通用网络模型和动态优先级调度策略,以最小化车联网中计算卸载任务的时延。

然而,目前关于这一领域的研究还相对有限,尤其是深度强化学习和拟牛顿法的结合使用,其效果和适用性还有待进一步探索。本文以基于深度强化学习和拟牛顿法在车联网的边缘计算为研究对象,探究其在车联网边缘计算环境中的应用效果和潜在价值;旨在结合深度强化学习和拟牛顿法,提出一种高效的卸载策略,以优化车联网边缘计算的总体性能。

本文的重要性在于,它能够车联网边缘计算中的任务卸载问题提供一种新的解决方案,从而提高车联网的任务卸载的效率和可靠性,促进其在智能交通、智能城市等领域的应用。同时,本文也将为未来相关领域的研究提供理论和实践的参考。

本文的主要贡献可以概括为以下2个方面。

1) 本文提出了拟牛顿法的深度强化学习双阶段在线卸载(QNRLO, Quasi-Newton deep reinforce-

ment learning dual-phase online offloading)算法,用于生成边缘计算环境下的最优任务卸载策略。该算法采取双阶段方法,初步利用批归一化的深度神经网络(DNN, deep neural network)形成初始卸载策略,并通过拟牛顿法对策略进行进一步细化,实现更接近最优解的策略搜索。这种方法增强了训练过程的稳定性,同时也降低了输入偏移和缩放因子的影响,为任务卸载问题提供了一个有效的解决方案。

2) 本文提出了一种非凸优化问题解决策略,通过引入拉格朗日算子和重构的对偶函数,成功将非凸问题转化为凸对偶的非凸优化问题,保证算法可达到全局最优解。同时,在优化模型性能时,本文充分考虑了车联网模型中系统传输时间的有效分配,进一步增强了模型的实用性和适用性。仿真结果验证了所提算法的优越收敛性能。

1 相关工作

深度学习在车联网的任务卸载决策优化方面展现了显著的潜力。文献[11]提出了一种基于软件定义信任的车辆临时网络架构。在该架构中,集中式的软件定义网络(SDN, software defined network)控制器扮演学习代理的角色,采用深度Q学习方法以获得最优的通信链路策略。文献[12]提出了一种抵抗具有先验知识的隐私攻击者的隐私导向任务卸载方法。首先定义了本地计算模型、信道模型和隐私损失模型,并用于量化隐私、时间和能耗等评估指标。其中,隐私损失被形式化为具有先验知识的攻击者成功攻击的概率。

强化学习在解决环境交互问题、适应性和泛化能力等方面有很大的优势。文献[13]探讨了在自动驾驶技术中应用强化学习的方法,并利用强化学习解决与环境的交互问题。强化学习使车辆能够通过与环境的持续交互学习,从而改进其决策。这一过程不需要依赖大量的标记数据,而是依靠从实际驾驶经验中积累的知识,这一点显著区别于其他需要大量预先标记好数据的机器学习技术。

深度强化学习结合了深度学习的特征表示能力和强化学习的决策能力,能够实现强大的端到端学习控制能力。文献[14]指出了这种方法不需要预先设计特征或手动制定决策规则,能够自动从环境反馈中学习状态与动作之间的映射关系,从而有效

地进行决策控制。深度强化学习的适应性强,不仅能够应对新的环境挑战,还能通过如迁移学习等策略,增强模型在不同任务中的泛化能力。虽然强化学习在决策控制领域表现出显著的能力,但其面临的一个关键挑战是如何提高样本效率,即如何在较少的交互次数内获得更高的学习效率。

为解决这一问题,需要探索样本效率更高的算法。拟牛顿法作为一种在优化问题求解中广泛应用的算法,提供了一个有价值的参考点。文献[15]提出了一种用于无约束随机优化问题的拟牛顿法。大多数现有文献假设随机优化问题中的有限和是期望的一个合理近似,因此可以设计一个拟牛顿法来利用公共随机数。文献[16]通过 Charnes-Cooper 转换来解决这一问题。文献[17]采用了基于对偶方法的凸优化问题解决方案,引入了拉格朗日乘子来构建对偶函数和对偶问题。

在神经网络研究中,批归一化技术的广泛应用得到了显著的关注。文献[18]引入了一种创新的领域泛化策略,这种策略依赖于批归一化统计数据,将来自各个领域的图像样本映射到同一潜在空间中。通过测量测试样本与每个领域累积统计数据的距离,推断出未知测试领域的特性。这种方法将批归一化统计数据积累在卷积层,以便有效地将图像样本映射到潜在空间。另一方面,文献[19]提出了一种解决深度神经网络训练过程中的梯度消失和梯度爆炸问题的方法,从而有效提升网络的训练速度和性能。通过对每个批次的输入数据进行归一化,网络中每一层的输入数据分布将变得更加稳定,这也有助于解决梯度消失和梯度爆炸问题。

本文针对车联网环境中的任务卸载决策优化问题,提出了一种结合深度强化学习和拟牛顿法的算法。该算法首先通过引入拉格朗日乘子和重构对偶函数,以时间因素为桥梁,建立本地计算和边缘计算之间的联系,旨在实现计算效率的最大化。接着,利用拟牛顿法精确找出各种卸载策略的最优解,并将这些解用于神经网络的深度强化学习过程中,以优化策略选择。此外,为了进一步提升网络训练的效率和稳定性,本文还采用了批归一化技术优化了神经网络的性能。

2 系统模型

本文在移动边缘计算 (MEC, mobile edge com-

puting) 的广泛框架下设计了一个包括 N 个车载单元 (OBU, on board unit) 以及一个 RSU 的模型,该模型可放置在框架中的任意区域,如图 1 所示,用户将任务卸载至 RSU 的 MEC 服务器进行处理,同时考虑了时延和能耗的影响。

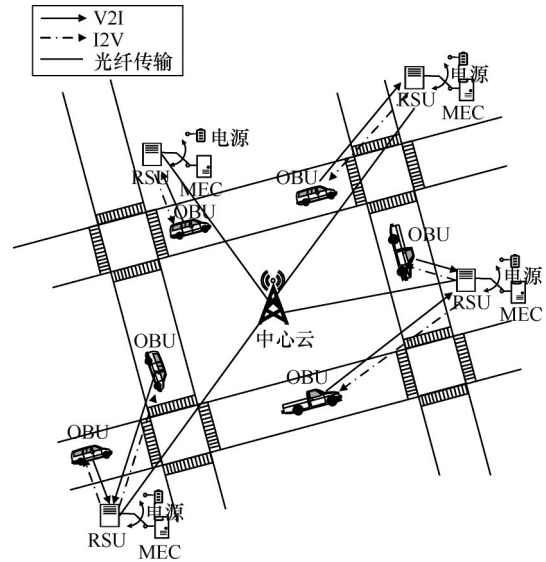


图1 基于移动边缘计算的车载单元与路侧单元任务卸载模型

在车联网任务卸载决策中,通过建立持续时间参数 aT 和 $\tau_i T$ 的关系,进一步计算得到 OBU 的本地计算率 R_i^{OBU} 与边缘计算率 R_i^{RSU} 。这 2 个计算率^[20]是决策卸载任务至关重要的衡量标准。具体地,若 R_i^{OBU} 优于 R_i^{RSU} ,则任务在 OBU 本地执行更为高效,否则,应考虑卸载至 RSU 以利用边缘计算资源。此策略确保了在动态的车联网环境中,每个任务都能以最优方式处理,有效提升了系统的整体性能和效率。

2.1 本地计算模型

本文在一个时间帧内考虑 OBU 和 RSU 的通信模式。在时间帧的初始部分,RSU 向所有的 OBU 广播^[21],提供最新的地图数据和能量供应。OBU 在规定的时间内卸载数据。车辆给 OBU 单元补充电量并计算,持续时间为 aT ,其中 $a \in [0,1]$,第 $i \in \mathcal{N} = \{1,2,\dots,N\}$ 个 OBU 补充的能量为

$$E_i^{\text{OBU}} = P_i aT \quad (1)$$

其中, P_i 表示第 i 个 OBU 的传输功率。根据文献[22],本地计算处理器计算率建模为 $f_i^3 = \frac{E_i^{\text{OBU}}}{k_i t_i}$,其中, f_i 表示处理器选择的计算率 (每秒周期数), $0 \leq t_i \leq T$ 表示 OBU 的计算时间, k_i 表示处理器芯片的计算

能量效率系数。根据文献[23], 本地计算率可表示为

$$R_i^{\text{OBU}} = \frac{f_i t_i}{\phi T} = \frac{(P_i a)^{\frac{1}{3}}}{\phi k_i^{\frac{1}{3}}} \quad (2)$$

其中, ϕ 表示处理一位原始数据所需的计算周期数, 且 $\phi > 0$ 。当 $t_i = T$ 时, R_i^{OBU} 达到最大。

2.2 边缘计算模型

边缘计算允许数据在离数据源更近的地方进行处理, 从而减少了时延并优化了资源的使用。在本文中, MEC服务器的加入不仅提高了整体系统的计算能力, 还通过智能任务卸载策略减少了车载单元之间的通信负担。此外, 本文还考虑了通信带宽、噪声功率等因素, 以实现更为精确的系统性能评估。

文献[24]指出, 在边缘计算中, MEC计算速度远快于传输速度, 故本文只考虑传输时间, 任务卸载中的时间分配模型如图2所示。其中, $\tau_i T$ 表示第 i 个 OBU 执行任务卸载的时间, $\tau_i \in [0, 1]$, 且 τ_i 和 a 的关系为

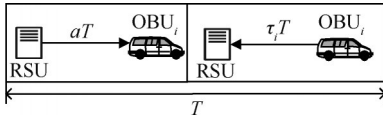


图2 任务卸载中的时间分配模型

$$\sum_{i=1}^N \tau_i + a \leq 1 \quad (3)$$

假设 P_i 为第 i 个 OBU 的传输功率, B 为通信带宽, N_0 为噪声功率, b 为通信开销, g_i 为第 i 个 OBU 与 RSU 之间的无线信道增益, 根据香农定理, 可以得到最大的传输容量, 即

$$B_i = \frac{B\tau_i T}{b} \text{lb} \left(1 + \frac{P_i g_i}{N_0} \right) \quad (4)$$

MEC 向 OBU 发射的能量为 E_i , 不考虑损耗, 最大的功率应该为 $P_i^* = \frac{E_i^{\text{OBU}}}{\tau_i T}$, 将 P_i^* 代入 B_i 中, 可以进一步得出边缘计算率 R_i^{RSU} , 即

$$R_i^{\text{RSU}} = \frac{B_i}{T} = \frac{B\tau_i}{b} \text{lb} \left(1 + \frac{P_i a g_i}{\tau_i N} \right) \quad (5)$$

3 问题分析

根据式(2)和式(5)中的所有系统参数, 假设在

所考虑的时间帧内无线信道增益 $\mathbf{g}_i = \{g_i | i \in \mathcal{N}\}$ 等其他参数已知, 那么在此时间帧内, MEC 模型的加权总计算速率可表示为

$$Q(\mathbf{x}_i, \mathbf{g}_i) \triangleq \sum_{i \in \mathcal{N}, j \in \mathcal{N}} (w_i x_i R_i^{\text{OBU}}(a) + w_j x_j R_i^{\text{RSU}}(a, \tau_i)) \quad (6)$$

其中, $\mathbf{x}_i = \{x_i, x_j | i, j \in \mathcal{N}\}$, $\tau = \{\tau_i | i \in \mathcal{N}\}$, w_i 为本地计算的权重, w_j 为边缘计算的权重。最佳计算率就是最大计算率, 相应的优化策略可以表示为问题 P1

$$\text{P1: } Q^*(\mathbf{x}_i, \mathbf{g}_i) = \max_{\tau, a} Q(\mathbf{x}_i, \mathbf{g}_i) \quad (7a)$$

$$\text{s.t. } \sum_{i=1}^N \tau_i + a \leq 1 \quad (7b)$$

$$a \geq 0, \tau_i \geq 0 \quad (7c)$$

$$\forall i \in \mathcal{N} \quad (7d)$$

$$\mathbf{x}_i = \{x_i | i \in \mathcal{N}\} \quad (7e)$$

通过 DNN 生成了卸载策略 \mathbf{x}_i , 因此可以将限制条件是非凸的 P1 问题转化成限制条件为凸的非凸优化问题 P2。证明过程详见附录 1。

$$\text{P2: } Q^*(\mathbf{g}_i) = \max_{\tau, a} Q(\mathbf{g}_i) \quad (8a)$$

$$\text{s.t. } \sum_{i=1}^N \tau_i + a \leq 1 \quad (8b)$$

$$a \geq 0, \tau_i \geq 0 \quad (8c)$$

$$\forall i \in \mathcal{N} \quad (8d)$$

针对车联网中的任务卸载决策问题, 本文采用了数学转化及优化算法来显著提升算法性能。通过引入拉格朗日算子和重构对偶函数, 将原先具有凸约束的非凸优化问题 P2 成功转化为凸优化问题。这一转化步骤的关键在于其不仅确保了求解过程能够稳定地找到全局最优解, 还显著简化了问题的结构。简化后的结构使得优化算法能够以更高效、更系统化的方式进行求解。其中, \mathcal{N} 为一个集合, 不需要对 \mathcal{N} 进行求偏导, 故只考虑 a 、 τ 和 λ , 然后通过求解拉格朗日函数的偏导数并令其等于零来找到最优解 P3。

$$\text{P3: } \max L(a, \tau, \lambda) = \sum_{i \in \mathcal{N}^{\text{OBU}}} w_i \varepsilon_1 \left(\frac{a}{k_i} \right)^{\frac{1}{3}} + \sum_{j \in \mathcal{N}^{\text{RSU}}} \left[w_j \varepsilon_2 \tau_j \ln \left(1 + \frac{\varepsilon_3 a}{\tau_j} \right) - \lambda \left(\sum_{j \in \mathcal{N}^{\text{RSU}}} \tau_j + a - 1 \right) \right] \quad (9)$$

其中, λ 是拉格朗日乘数, $\varepsilon_1 = \frac{(P_i)^{\frac{1}{3}}}{\phi}$, $\varepsilon_2 = \frac{B}{b \ln 2}$,

$\varepsilon_3 = \frac{P_i g_i}{N}$ 。令 $\frac{\partial L}{\partial \tau_j} = 0$ 和 $\frac{\partial L}{\partial a} = 0$ ，根据计算可以得到 $\{\tau_j, a\}$ 最优解的关系式为

$$\frac{\tau_j^*}{a^*} = \frac{\varepsilon_2}{f(\lambda)}, \quad \forall j \in \mathcal{N} \quad (10)$$

$$\tau_j^* = \varepsilon_3 a^* f(\lambda), \quad \forall j \in \mathcal{N} \quad (11)$$

$$a^* = \frac{1}{1 + \left(\sum_{j \in \mathcal{N}_{\text{RSU}}} \frac{\varepsilon_3}{f(\lambda)} \right)} \triangleq g(\lambda) \quad (12)$$

其中, $f(\lambda) = - \left[W \left(- \frac{1}{\exp \left(1 + \frac{\lambda}{w_j \varepsilon_2} \right)} \right) \right]^{-1} - 1$,

$W(x)$ 表示 Lambert-W 函数, $f(\lambda)$ 的证明由文献[25]得到。根据式(11)和式(12), 可以把问题P3转化成对偶函数 $D(\lambda)$ 的问题P4, 且对偶函数 $D(\lambda)$ 为凸函数, 证明过程详见附录2。

$$P4: D(\lambda) = \inf_{\lambda \geq 0} \max_{a, \tau} L(a, \tau, \lambda) \quad (13)$$

4 QNRLO 算法

QNRLO 算法先通过把非凸优化问题P1转化成具有凸对偶的非凸优化问题P4, 再通过拟牛顿法找到最优解 λ^* 。强化学习算法如算法1所示。

算法1 强化学习算法

输入 信道增益 g_t

输出 最佳卸载决策 \mathbf{x}_t^*

- 1) 初始化深度神经网络。
- 2) 定义网络前向传播。

- 3) 根据神经网络输出进行量化。
- 4) 生成 k 种卸载策略 $\mathbf{x}_{t,k}$ 。
- 5) 调用算法2, 并返回最优卸载策略 \mathbf{x}_t^* 。

4.1 算法简介

QNRLO 算法框架如图3所示, 其设计的核心在于选用DNN作为本文解决方案的主要工具。选择DNN的原因主要基于以下几点[26]: 首先, DNN因其卓越的表达能力而突出, 能够处理复杂的非线性关系, 并且通过学习输入数据中的高层次特征, 显著提高分类和回归任务的准确性; 其次, DNN的自适应性使得它能够自动提取并表示输入数据的特征, 大幅简化了特征工程的复杂度; 最后, DNN支持端到端训练, 通过反向传播算法, 避免了传统机器学习方法中需手动进行的特征提取和分类器设计等烦琐步骤。正是由于这些优势, DNN成为在边缘服务器端部署的理想选择, 以应对快速增长的数据处理需求。

本文算法由2个交替的阶段组成: 卸载动作生成和卸载策略更新。卸载动作生成基于一个四层的DNN。在这个DNN中, 第一层为输入层, 中间两层为隐藏层, 最后一层为输出层。隐藏层的每层都有其独特的权重和偏置进行信息的线性转换, 这些权重和偏置在训练过程中进行动态调整, 以适应不断变化的数据。同时, 加入批归一化将输入数据的分布近似调整为标准正态分布, 进而降低输入偏移和缩放因子的影响, 提高了网络的训练速度和稳定性。

$$\hat{\mathbf{x}}_i = \frac{\mathbf{g}_i - m_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad (14a)$$

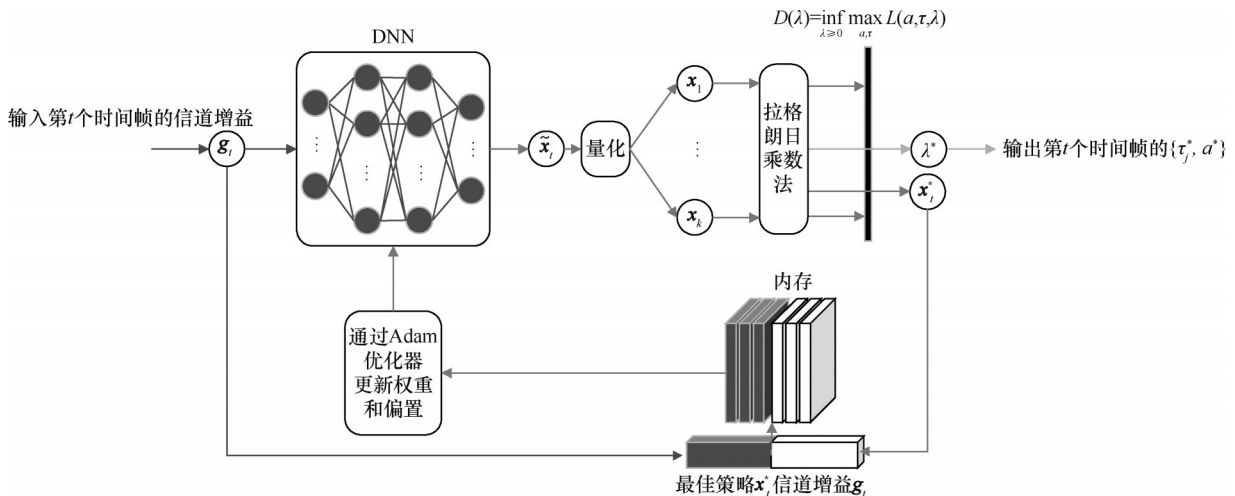


图3 QNRLO 算法框架

$$\sigma_R^2 = \mu\sigma_R^2 + (1 - \mu)\sigma_B^2 \quad (14b)$$

$$\mu_R = \mu\mu_R + (1 - \mu)\mu_B \quad (14c)$$

其中, \mathbf{g}_t 为输入特征值, m_B 为批样本数, μ_B 为批样本均值, σ_B^2 为批样本方差, ε 为防零除因子, \hat{x}_i 为归一化后的特征值, μ_R 为运行均值, σ_R^2 为运行方差, μ 为动量。

输出层也具有相应的权重和偏置, 用于生成最终的卸载动作。在第 t 个时间帧中, DNN 以信道增益 \mathbf{g}_t 为输入, 输出一组预测概率 $\tilde{\mathbf{x}}_t = \{\tilde{x}_{t,i} | \tilde{x}_{t,i} \in (0,1), i = 1, 2, \dots, N\}$, 其中, $\tilde{x}_{t,i}$ 是 $\tilde{\mathbf{x}}_t$ 的第 i 个元素。然后基于问题 P4 求得的公式选择一个最佳动作 \mathbf{x}_t^* 。根据 λ^* 和式(11)、式(12)相应的 $\{\tau_j^*, a^*\}$ 作为 \mathbf{g}_t 的解被输出, 确保满足式(8b)~式(8d)中列出的所有物理约束。通过任务卸载策略 \mathbf{x}_t , 比较奖励 $Q(\mathbf{g}_t, \mathbf{x}_t)$, 并将获得的最优任务卸载策略 \mathbf{x}_t^* 和信道增益 \mathbf{g}_t 加入内存进行记录。随后, 在第 t 个时间帧的策略更新阶段, 从内存中提取一批训练样本来训练 DNN, 得到新的权重和偏置。在接下来的时间帧中, 新的权重和偏置将用于处理观察到的新信道增益 \mathbf{g}_{t+1} 和卸载决策 \mathbf{x}_{t+1} 。此迭代不断重复, DNN 中的隐藏层权重和偏置逐渐得到改进。

4.2 卸载策略的生成

假设在第 t 个时间帧, 信道增益为 \mathbf{g}_t , 其中, $t = 1, 2, \dots, N$ 。在 $t = 1$ 时, DNN 的权重和偏置进行了随机初始化, 并且其遵循零均值正态分布。

根据万能近似定理, 在神经元上使用如 sigmoid、ReLU 和 tanh 等适当的激活函数时, 一个隐藏层带有足够神经元就可以逼近任何连续映射。于是, 在本文的模型中, 隐藏层使用了 Leaky ReLU 激活函数^[27], 即

$$f(\theta) = \begin{cases} x, & \theta > 0 \\ \alpha\theta, & \text{其他} \end{cases} \quad (15)$$

本文使用 Leaky ReLU 激活函数试图解决神经元“死亡”的问题。Leaky ReLU 激活函数通过允许负输入有一个小的非零输出, 有助于防止神经元完全不活跃, 从而改善模型性能和稳定性。

在输出层, 本文选择使用 sigmoid 激活函数 $f(\theta) = \frac{1}{1 + e^{-\theta}}$, 这样可以确保松弛卸载动作 $\tilde{x}_{t,i}$ 的值在(0,1)范围内。

为了从 $\tilde{\mathbf{x}}_t$ 获取 K 个二进制卸载动作, 本文进行

了量化操作, 即 $\mathbf{x}_k \in \{0,1\}^{(N)}$, 其中, K 是设计中的一个参数, 生成 k 个量化策略; N 是 OBU 的数量。下面介绍本文的量化方案。

第一个量化策略通过以下方式获得

$$x_{k,i} = \begin{cases} 1, & \hat{x}_{t,i} > 0.5 \\ 0, & \hat{x}_{t,i} \leq 0.5 \end{cases} \quad (16)$$

其中, $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$ 。

本文剩下的 $k - 1$ 个量化策略通过比较 $\tilde{\mathbf{x}}_t$ 和 0.5 的大小产生。通过以下方法进行重新排序: $|\tilde{x}_t^{(1)} - 0.5|, |\tilde{x}_t^{(2)} - 0.5|, \dots, |\tilde{x}_t^{(i)} - 0.5|, \dots, |\tilde{x}_t^{(N)} - 0.5|$, $\tilde{x}_t^{(i)}$ 表示第 t 个时间帧第 i 个排序后的数据统计量。量化策略按照如下方法进行量化

$$x_{k,i} = \begin{cases} 1, & \tilde{x}_{t,i} \geq \tilde{x}_t^{(k)} \\ 0, & \tilde{x}_{t,i} < \tilde{x}_t^{(k)} \end{cases} \quad (17)$$

$$\mathbf{x}_t^* = \max_{\mathbf{x}_k} Q^*(\mathbf{g}_t, \mathbf{x}_k) \quad (18)$$

4.3 最优解算法

通过最优解算法, 从量化后的卸载策略 $\mathbf{x}_k = \{x_{k,i} | x_{k,i} \in [0,1], i = 1, 2, \dots, N\}$ 中选择最优卸载策略, 解决分配问题 (问题 P4)。为了高效地找到最优解, 本文采用了基于拟牛顿法的优化方法, 如 L-BFGS (limited-memory broyden-fletcher-goldfarb-shanno)。其作为一种高效的迭代优化手段, 致力于实现近似牛顿法的优化效果, 同时避免了直接计算 Hessian 矩阵所带来的高计算成本。

相较于传统依赖一阶优化技术, 如梯度下降法, 拟牛顿法显示出了其在资源分配问题求解上的显著优势。特别是, 它的超线性甚至二次的收敛速度, 大大减少了达到最优解所需的迭代次数。这一特性不仅显著提高了算法的效率, 也在处理具有复杂或非凸目标函数的优化问题时展现了独特的优势。

此外, 拟牛顿法的应用在实际的车联网场景中尤其重要。在这些场景下, 资源分配问题往往因为网络条件的动态变化和任务的多样性而显得尤其复杂。拟牛顿法的高效性使算法能够快速适应这些变化, 确保能够及时且准确地做出资源分配决策, 从而优化整个车联网系统的性能。

在拟牛顿法的每次迭代中, 需要调用目标函数来评估目标函数的值。如果 OBU 数量为 N , 则目标函数在整个优化过程中总的时间复杂度是 $O(N)$, 空间复杂度是 $O(N + d \maxcor)$, 其中, d 是维数, \maxcor 是 Hessian 矩阵更新向量数量的参

数, 这部分的空间需求相对固定。由于问题是一维的, 其影响主要通过 N 来体现, 故空间复杂度和 OBU 数量呈线性关系, 进一步简化为主要由 N 决定的 $O(N + \max\text{cor})$ 。因为算法的时间复杂度和空间复杂度都与 OBU 数量 N 呈线性关系, 所以算法在资源受限的边缘计算环境中是可行的。

基于拟牛顿法的资源分配算法如算法 2 所示。随着迭代的进行, P4 的目标函数值单调增加。由于 P4 的目标函数是凸函数, 因此拟牛顿法可以保证收敛。

算法 2 基于拟牛顿法的资源分配算法

输入 量化参数 x_i

输出 最佳卸载率 R_i^{opt}

1) 初始化参数和方程式。

2) 对卸载策略 x 进行分解: 0 为本地计算, 1 为边缘计算。

3) 采用拟牛顿法, 根据式(13)的对偶函数 $D(\lambda)$ 获得对应卸载策略 x 的最优分配时间。

4) 根据式(2), 计算本地计算率并记为 R_i^{OBU} 。

5) 根据式(5), 计算边缘计算率并记为 R_i^{RSU} 。

6) 比较同一时间生成的不同卸载策略 R_i^{opt} , 输出最大值。

利用不同方法寻求最优策略参数, 可以改进强化学习的收敛性能^[28]。为此, 本文进一步讨论拟牛顿法的优势。图 4 比较了 4 种方法(拟牛顿法、牛顿法、Powell 法和梯度下降法)的归一化计算率。作为牛顿法的一种改进, 拟牛顿法避免了直接计算 Hessian 矩阵及其逆, 通过近似的方式更新 Hessian 的逆, 因而可以更快地收敛, 同时减少了计算成本。牛顿法可以提供二次收敛速度, 这意味着它在接近最优解时收敛非常快。Powell 法是一种不需要计算梯度信息的优化方法, 导致收敛速度慢。梯度下降法在面对复杂函数的时候, 需要迭代更多次才能达到收敛。

5 仿真分析

本节使用计算机模拟仿真来评估所提 QNRLO 算法的性能。在所有的模拟仿真中, 假设 OBU 充电功率为 $P=3\text{ W}$ 。

本文模拟参考文献[29-30]进行建模, 由每个边缘服务器独立训练, 而云节点需要聚合不同边缘服务器更新生成全局模型。边缘服务器训练所需要的

数据(信道增益)由代码随机生成, 验证集使用的是 DROO (deep reinforcement learning-based online offloading) 算法^[31]作为基准。文献[32]提供了一种方案, 可以解决实际模型训练的数据采集, 下一代基站 (gNB, generation node B) 内部署的边缘计算服务器负责执行信道频率响应 (CFR, channel frequency response) 的存储, 然后利用这些数据训练深度学习信道估计模型。通过这种方式, 边缘计算服务器能够生成一个经过训练的预测模型, 随后这个模型被传输给 OBU 和 RSU, 使得这些设备能完成信道估计的任务。本文主要在信道增益已获得的前提下分析如何实现最优分配, 不讨论在实际应用中采集信道增益所涉及的通信量和计算量问题。

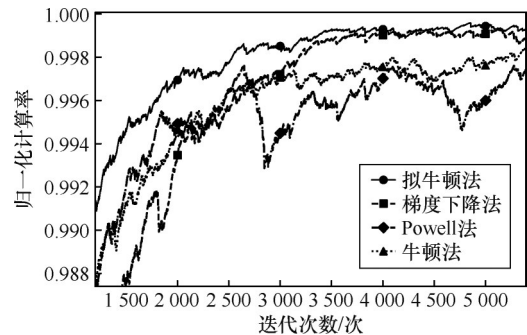


图 4 4 种方法的归一化计算率

第 i 个 RSU 在时间帧 t 的平均信道增益记为 \bar{g}_i 。

$$\bar{g}_i = A_d \left(\frac{3 \times 10^8}{4\pi f_c d_i} \right)^{d_e} \text{ 遵循自由空间传输损耗模型, 其}$$

中, $A_d = 4.11$ 表示天线增益, $f_c = 915\text{ MHz}$ 表示载波频率, d_i 表示传输距离, $d_e = 2.8$ 表示传输损耗指数。信道增益在一个时间帧内被假定为保持不变, 并且在一个时间帧到另一个时间帧之间独立变化。

所有的模拟仿真都在一台服务器上执行, 该服务器配备有处理器为 12th Gen Intel Core i5-12500H 的 CPU 和 16 GB 内存。

本文在所提 QNRLO 算法中使用 DNN, 包括一个输入层、两个隐藏层和一个输出层。第一个隐藏层有 256 个神经元, 第二个隐藏层有 128 个神经元。在隐藏层中加入批归一化, 动量为 0.9, 防零除因子为 1×10^{-5} 。本文发现, 一个简单的两层感知器已足以实现令人满意的收敛性能, 但通过进一步优化 DNN 参数, 可以获得更好的收敛性能。

本文使用 TensorFlow 2.12 在 Python 中实现了 QNRLO 算法, 并将训练间隔设置为 $t=10$, 训练批

量大小为 256, 内存大小为 1 024, Adam 优化器的学习率为 0.01。

首先, 本文考虑一个具有 10 个 OBU 的 MEC 网络。定义归一化计算速率 $Q^*(g_t) = \frac{Q(g_t)}{\max Q(g_t)}$, 取值范围为 [0,1], 其中, $\max Q$ 为测试集的最优解。

本文使用的损失函数是二进制交叉熵损失函数 $L(y, p) = -\sum_{i=1}^N y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$ (19) 其中, N 是样本数量, y 是真实标签, p 是预测概率。

当 $N=10$ 和 $K=10$ 时, OBU 的 QNRLO 算法的训练损失率和归一化计算率如图 5 所示。在 2 000 次迭代时, 损失函数收敛至 0.09。算法在学习过程中通过强化学习不断调整其决策策略, 以适应环境的变化, 从而导致损失函数出现轻微的波动。即便如此, 损失函数最终仍能继续收敛, 稳定在大约 0.05 的水平。

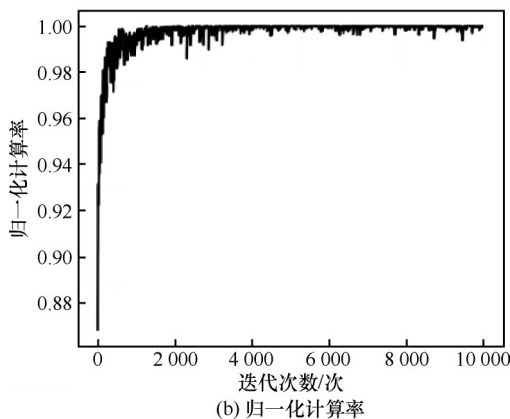
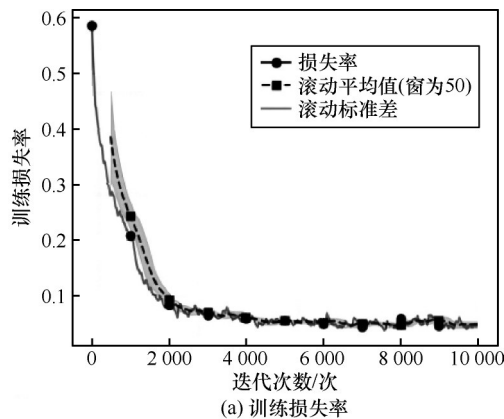


图 5 当 $N=10$ 和 $K=10$ 时, OBU 的 QNRLO 算法的训练损失率和归一化计算率

从图 5(a)中可以看到, 随着迭代的进行, 滚动平均值趋于平稳, 并在最后阶段收敛到一个近似恒定的值。灰色的方差区域也在缩小, 这表明数据的

变化在减少。

图 5(b)中同样是在 2 000 次迭代时达到收敛, 在每次权重变换时, 可以观察到计算率中出现了抖动, 同样也很快恢复收敛, 最终归一化计算率收敛于 1。

本文进一步研究了不同算法参数对神经网络收敛性能的影响, 包括不同的内存、批量和学习率。图 6 展示了神经网络内存大小分析。在不同内存下, 系统均实现了迅速收敛。但是, 从图 7 可以看到, 当内存 (即记录个数) 为 128 时, 标准差为 0.016; 当内存为 1 024 和 2 048 时, 标准差都为 0.009。这表明虽然增大内存能够略微降低标准差, 但增加到 2 048 会显著增加计算负担和训练时间。基于这些观察, 本文将内存设为 1 024。

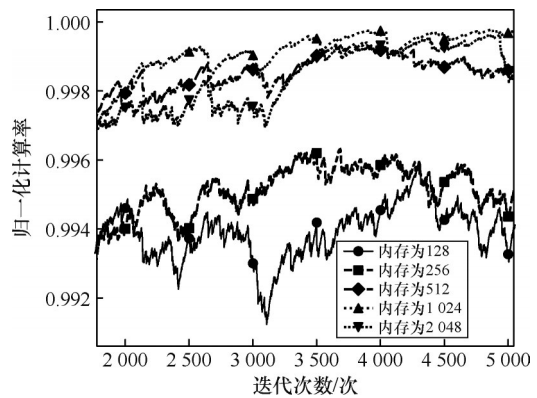


图 6 神经网络内存大小分析

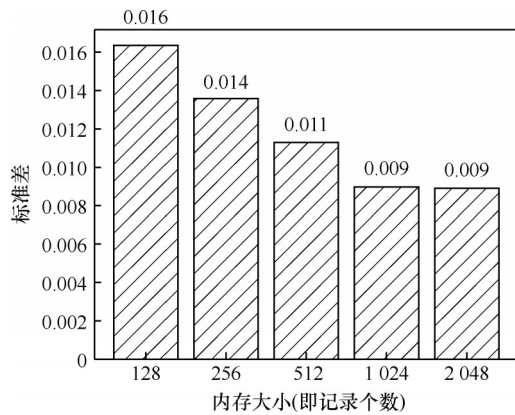


图 7 神经网络内存大小标准差分析

图 8 展示了神经网络批量大小分析。结果显示, 批量为 32 时的收敛性明显较差, 可能受到噪声的干扰, 而其他批量大小均表现良好。然而, 如图 9 所示, 当批量为 256 时, 进一步增大批量并不会使数据的稳定性提升。在引入批归一化后, 当批

量为 256、512 和 1 024 时，每个用户的平均处理时间均为 0.14 个时间帧。为了减少计算量，本文在后续实验中将批量设为 256。

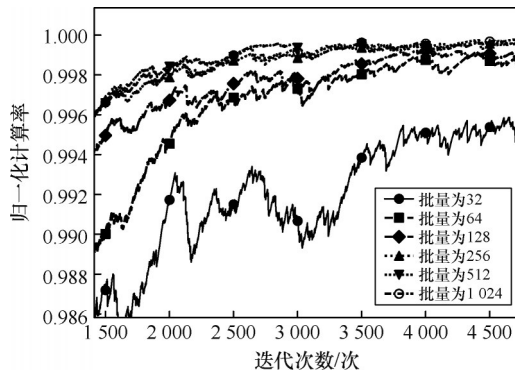


图 8 神经网络批量大小分析

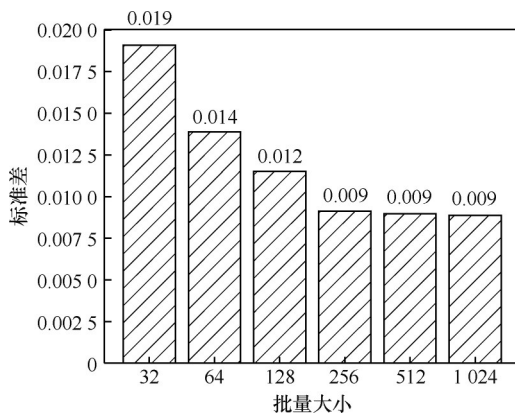


图 9 神经网络批量大小标准差分析

图 10 展示了神经网络学习率大小分析。从图 10 中可以很清晰地观察到，学习率为 0.001 时收敛速度明显下降。学习率为 0.01 和 0.1 的曲线基本一样，但是在学习率为 0.01 时，平均每个用户训练时间为 0.24 个时间帧，而学习率为 0.1 时，平均每个用户训练时间为 0.14 个时间帧。故本文在后续的实验中设置学习率为 0.1。

本文在相同参数下比较了 QNRLO 算法与 Async-Idd (asynchronous federated learning with identically distributed data) [33] 和 HL (hybrid learning) [34] 算法的归一化计算率及平均值和标准差，分别如图 11 和图 12 所示。从图 11 中可以看到，3 种算法都在 2 000 次迭代时达到了收敛，这说明它们在本次实验设定下具有相似的收敛速度。从图 12 中可以看到，QNRLO 算法的归一化计算率更稳定，标准差为 0.008 9，平均值为 0.997 9。综上所述，本文算法具有良好的收敛速度和稳定性。

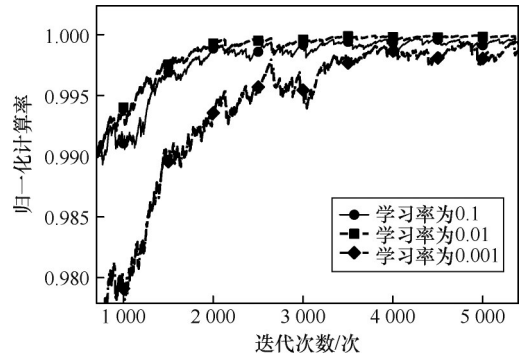


图 10 神经网络学习率大小分析

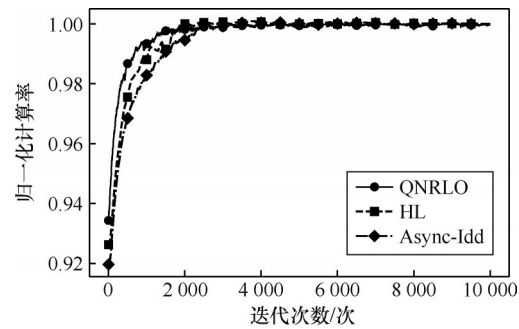


图 11 3 种算法的归一化计算率

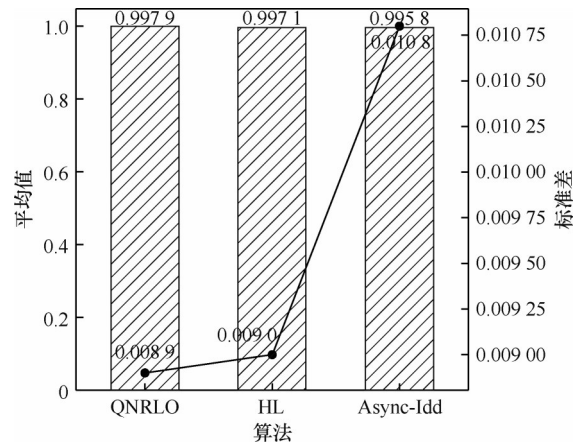


图 12 3 种算法的平均值和标准差

6 结束语

本文提出了一种基于深度学习的 QNRLO 算法，用于 MEC 网络中的计算任务卸载。通过计算机仿真模拟实验，证明了该算法在不同场景和参数设置下都展示了出色的收敛性能。首先，实验结果表明，在迭代了 2 000 次后，QNRLO 算法的损失函数和归一化计算率均达到了收敛，说明算法具有很高的稳定性和可靠性。其次，本文还研究了不同算法参数设置对神经网络性能的影响，如内存、批量和学习率等，结果显示，在不牺牲收敛性能的前提

下,一些参数的适当调整可以进一步优化计算负担和训练时间。值得强调的是,本文提出的QNRLO算法与传统的DROO算法相比表现出了更加优越的性能,这一结论已通过实验得到了验证。

尽管QNRLO算法表现出色,但也有一些局限性。实验主要是在模拟环境中进行的,未来可能需要在真实场景中进一步验证算法的有效性。未来工作可能会集中在以下几个方面:进一步优化算法的性能,扩展应用场景,以及考虑更多的性能指标。

附录1 P2问题为具有凸对偶的非凸优化问题的证明

P2问题是一个求最大值问题,已知 $\max_{\tau, a} Q(\mathbf{x}, \tau, a)$ 是一个非凸函数,但是根据附录2,该函数的对偶函数是一个凸函数,为了满足对偶函数的最优解是P2问题的最优解,需要验证两者是否存在强对偶性。

线性约束条件 $\sum_{i=1}^N \tau_i + a \leq 1$ 是凸的,定义了一个凸集。

非负约束 $a \geq 0, \tau_i \geq 0$ 也是凸的,同样是一个凸集。本文取 $a = \frac{1}{2N+2}$ 和 $\tau_i = \frac{1}{2N+2}$,满足上述不等式均成立。故满足Slater条件。

对偶函数是凸函数,如附录2所示,所以P2问题和对偶函数之间存在强对偶性。通过多次实验验证,将所得 λ^* 代入式(11)和式(12),满足KKT(Karush-Kuhn-Tucker)条件,故此最优解是全局最优解。综上所述,对偶函数的最优解就是目标函数的全局最优解。证毕。

附录2 式(13)是凸函数的证明

本文通过求 $L(a, \tau_j, \lambda)$ 对 a 和 τ_j 的偏导来证明对偶函数 $D(\lambda)$ 为凸函数。

a 的二阶偏导数为

$$L''_{aa} = -\frac{2\varepsilon_1 w_i}{9k_i^2 \left(\frac{a}{k_i}\right)^{\frac{5}{3}}} - \frac{\varepsilon_2 \varepsilon_3^2 w_j}{\tau_j \left(\frac{a\varepsilon_3}{\tau_j} + 1\right)^2} \quad (20)$$

τ_j 的二阶偏导数为

$$L''_{\tau_j \tau_j} = -\frac{a^2 \varepsilon_2 \varepsilon_3^2 w_j}{\tau_j^3 \left(\frac{a\varepsilon_3}{\tau_j} + 1\right)^2} \quad (21)$$

由式(20)和式(21)可知, $L(a, \tau_j, \lambda)$ 是凹函数,故 $D(\lambda)$ 为凸函数。证毕。

参考文献:

- [1] SONKOLY B, CZENTYE J, SZALAY M, et al. Survey on placement methods in the edge and beyond[J]. IEEE Communications Surveys & Tutorials, 2021, 23(4): 2590-2629.
- [2] REN J, ZHANG D Y, HE S W, et al. A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet[J]. ACM Computing Surveys, 2019, 52(6): 125.
- [3] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: the communication perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.
- [4] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [5] YOUSEFPOUR A, FUNG C, NGUYEN T, et al. All one needs to know about fog computing and related edge computing paradigms: a complete survey[J]. Journal of Systems Architecture, 2019, 98: 289-330.
- [6] ABDEL-HALIM I T, FAHMY H M A. Mobility prediction in vehicular ad-hoc networks: prediction aims, techniques, use cases, and research challenges[J]. IEEE Intelligent Transportation Systems Magazine, 2021, 13(2): 105-126.
- [7] GUPTA M, BENSON J, PATWA F, et al. Secure V2V and V2I communication in intelligent transportation using cloudlets[J]. IEEE Transactions on Services Computing, 2022, 15(4): 1912-1925.
- [8] NING Z L, ZHANG K Y, WANG X J, et al. Intelligent edge computing in Internet of vehicles: a joint computation offloading and caching solution[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(4): 2212-2225.
- [9] LUO Q Y, LI C L, LUAN T H, et al. Self-learning based computation offloading for Internet of vehicles: model and algorithm[J]. IEEE Transactions on Wireless Communications, 2021, 20(9): 5913-5925.
- [10] BOZORGCHENANI A, MAGHSUDI S, TARCHI D, et al. Computation offloading in heterogeneous vehicular edge networks: on-line and off-policy bandit solutions[J]. IEEE Transactions on Mobile Computing, 2022, 21(12): 4233-4248.
- [11] ZHANG D J, YU F R, YANG R Z, et al. Software-defined vehicular networks with trust management: a deep reinforcement learning approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(2): 1400-1414.
- [12] GAO H H, HUANG W Q, LIU T, et al. PPO2: location privacy-oriented task offloading to edge computing using reinforcement learning for intelligent autonomous transport systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(7): 7599-7612.
- [13] KIRAN B R, SOBH I, TALPAERT V, et al. Deep reinforcement learning for autonomous driving: a survey[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(6): 4909-4926.
- [14] WANG X, WANG S, LIANG X X, et al. Deep reinforcement learning: a survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(4): 5064-5078.
- [15] MENICKELLY M, WILD S M, XIE M. A stochastic quasi-newton method in the absence of common random numbers[J]. arXiv Preprint, arXiv: 2302.09128, 2023.
- [16] HONG T, LIU R, LIU Z W, et al. An asynchronous collision-tolerant

- ACRDA scheme based on satellite-selection collaboration-beamforming for LEO satellite IoT networks[J]. *Sensors*, 2023, 23(7): 3549.
- [17] KRUTIKOV V, TOVBIS E, BYKOV A, et al. Properties of the quadratic transformation of dual variables[J]. *Algorithms*, 2023, 16(3): 148.
- [18] SEGU M, TONIONI A, TOMBARI F. Batch normalization embeddings for deep domain generalization[J]. *Pattern Recognition*, 2023, 135: 109115.
- [19] MA Y T, KLABJAN D. Diminishing batch normalization[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2024, 35(5): 6544-6557.
- [20] WANG F. Computation rate maximization for wireless powered mobile edge computing[C]//*Proceedings of the 2017 23rd Asia-Pacific Conference on Communications (APCC)*. Piscataway: IEEE Press, 2017: 1-6.
- [21] QIAN Y F, JIANG Y Y, HU L, et al. Blockchain-based privacy-aware content caching in cognitive Internet of vehicles[J]. *IEEE Network*, 2020, 34(2): 46-51.
- [22] GUO S T, XIAO B, YANG Y Y, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C]//*Proceedings of the IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. Piscataway: IEEE Press, 2016: 1-9.
- [23] WANG Y T, SHENG M, WANG X J, et al. Mobile-edge computing: partial computation offloading using dynamic voltage scaling[J]. *IEEE Transactions on Communications*, 2016, 64(10): 4268-4282.
- [24] YOU C S, HUANG K B, CHAE H. Energy efficient mobile cloud computing powered by wireless energy transfer[J]. *IEEE Journal on Selected Areas in Communications*, 2016, 34(5): 1757-1771.
- [25] BI S Z, ZHANG Y J. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. *IEEE Transactions on Wireless Communications*, 2018, 17(6): 4177-4190.
- [26] YANG B, CAO X L, XIONG K, et al. Edge intelligence for autonomous driving in 6G wireless system: design challenges and solutions[J]. *IEEE Wireless Communications*, 2021, 28(2): 40-47.
- [27] LI C L, RAVANBAKHSH S, POCZOS B. Annealing Gaussian into ReLU: a new sampling strategy for leaky-ReLU RBM[J]. *arXiv Preprint*, arXiv: 1611.03879, 2016.
- [28] MANGANINI G, FIORAVANTI S, RAMPONI G. Newton-based policy search for networked multi-agent reinforcement learning[C]//*Proceedings of the 2022 IEEE 61st Conference on Decision and Control (CDC)*. Piscataway: IEEE Press, 2022: 7241-7247.
- [29] LI P Z, SEFEROGLU H, DASARI V R, et al. Model-distributed DNN training for memory-constrained edge computing devices[C]//*Proceedings of the 2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. Piscataway: IEEE Press, 2021: 1-6.
- [30] LI Y J, ZENG Z B, LI J, et al. Distributed model training based on data parallelism in edge computing-enabled elastic optical networks[J]. *IEEE Communications Letters*, 2021, 25(4): 1241-1244.
- [31] HUANG L, BI S Z, ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. *IEEE Transactions on Mobile Computing*, 2020, 19(11): 2581-2593.
- [32] LIAO Y, CAI Z R, SUN G D, et al. Deep learning channel estimation based on edge intelligence for NR-V2I[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, 23(8): 13306-13315.
- [33] YANG J L, DUAN Y X, QIAO T, et al. Prototyping federated learning on edge computing systems[J]. *Frontiers of Computer Science*, 2020, 14(6): 146318.
- [34] FENG C Y, YANG H H, WANG S Y, et al. Hybrid learning: when centralized learning meets federated learning in the mobile edge computing systems[J]. *IEEE Transactions on Communications*, 2023, 71(12): 7008-7022.

[作者简介]



章坚武 (1961-), 男, 浙江杭州人, 博士, 杭州电子科技大学教授、博士生导师, 主要研究方向为移动通信、多媒体信号处理与人工智能、通信网络与信息安全等。



芦泽韬 (2000-), 男, 江西九江人, 杭州电子科技大学硕士生, 主要研究方向为边缘计算、强化学习等。



章谦骅 (1990-), 男, 浙江杭州人, 之江实验室天基计算研究中心工程师, 浙江大学博士生, 主要研究方向为天基计算、激光通信、计算卸载等。



詹明 (1975-), 男, 河南新县人, 博士, 台州学院教授、博士生导师, 主要研究方向为信道编码理论与技术、工业无线传感器网络、高可靠低时延通信和安全通信技术。